Machine Learning
Homework Assignment 3

Nishika Chopra Karanpreet Singh Wadhwa	nc2259 N18598308 ksw352 N13337853

NC2259 – Nishika Chopra
KSW352 – Karanpreet Singh Wadhwa

ASS-3 Irag

(5)

$$\begin{bmatrix} 0 & 14 \\ 6 & 9 \end{bmatrix}$$

(a) $Av = \lambda v$

$$\begin{bmatrix} 0-\lambda & 14 \\ 6 & 9-\lambda \end{bmatrix} = \begin{bmatrix} -\lambda & 14 \\ 6 & 9-\lambda \end{bmatrix}$$

$$-\lambda(9-\lambda) - (14 \times 6) = -9\lambda + \lambda^2 - 84$$

$$= \lambda^2 - 9\lambda - 84$$

(b) $\lambda^2 - 9\lambda - 84 = 0$

$\lambda = 14.71$ or $\lambda = -5.71$

(c) For $\lambda = 14.71$

$$\begin{bmatrix} -14.71 & 14 \\ 6 & 9-14.71 \end{bmatrix} = \begin{bmatrix} -14.71 & 14 \\ 6 & -5.71 \end{bmatrix}$$

$-14.71x + 14y = 0$ & $\sqrt{x^2+y^2} = 1$

$6x - 5.71y = 0$

Solving for ( )

c) $\Rightarrow K_1 = 14.71 \qquad K_2 = -5.71$

for K_1 $\begin{bmatrix} -14.71 & 14 \\ 6 & -5.71 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$

~~for K~~ $6x - 5.71y = 0 \qquad \sqrt{x^2+y^2} = 1$

$x = \frac{5.71y}{6}$

$= 0.952y \qquad \sqrt{0.9063y^2 + y^2} = 1$

$y = 0.7243$ & $x = 0.6895$ eigen vector $[0.6895, 0.7243]$

for K_2 $\begin{bmatrix} 5.71 & 14 \\ 6 & 14.71 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$

$\Rightarrow 6x + 14.71y = 0, \quad \sqrt{x^2+y^2} = 1$

$x = -\frac{14.71}{6} y$

~~[illegible]~~ ~~, 2.648~~ eigen vector $[0.92594, 0.37762]$

$K_1 \qquad K_2$

d) eigen values = $[-5.7102\,8893, 14.7102\,8893]$

eigen vector = $\begin{bmatrix} -0.9259401 & -0.6894021 \\ 0.37767039 & -0.72437887 \end{bmatrix}$

~~eck~~ eigen vector for $\lambda 1$: $[-0.9259401 \quad 0.37767039]$

eigen vector for $K2$: $[-0.6894021 \quad -0.72437887]$

(3a)

$$\begin{bmatrix} 5 & 2 & 4 \\ 9 & 6 & 4 \\ 7 & 1 & 0 \\ 2 & 5 & 6 \end{bmatrix}$$

$S_1 = \frac{5+9+7+2}{4}$

$= 5.75$

$S_2 = \frac{2+6+1+5}{4} = \frac{14}{4} = 3.5$

$$S_3 = \frac{14}{4} = 3.5$$

$$B = \begin{bmatrix} -0.75 & -1.5 & 0.5 \\ 3.25 & 2.5 & 0.5 \\ 1.25 & -2.5 & -3.5 \\ -3.75 & 1.5 & 2.5 \end{bmatrix}$$

(b) $$S_{1,3} = \sum_t \frac{(x_i^t - s_i) * (x_j^t - s_j)}{N-1}$$

$$= \frac{(x_1^1 - s_1) * (x_3^1 - s_3) + (x_1^2 - s_1) * (x_3^2 - s_3) + (x_1^3 - s_1)(x_3^3 - s_3) * (x_1^4 - s_1)}{N-1}$$

$$= \frac{(-0.75)(0.5) + (3.25)(0.5) + (1.25)(-3.5) + (-3.75)(2.5)}{3}$$

$$= \frac{12.5}{3} = 4.166\overline{6}$$

(c) Sample Covariance Matrix of B

$$\begin{bmatrix} 8.9166667 & 0.166\overline{6} & -4.166\overline{6} \\ 0.166\overline{6} & 5.666\overline{6} & 4.333\overline{3} \\ -4.166\overline{6} & 4.333\overline{3} & 6.333\overline{3} \end{bmatrix}$$ 3×3

~~(.75 - 5.75)(2.5 - 3.75)~~ Eigenvalue for this

$$\begin{bmatrix} 0.99703343 & 12.97881887 & 7.94081437 \end{bmatrix}$$

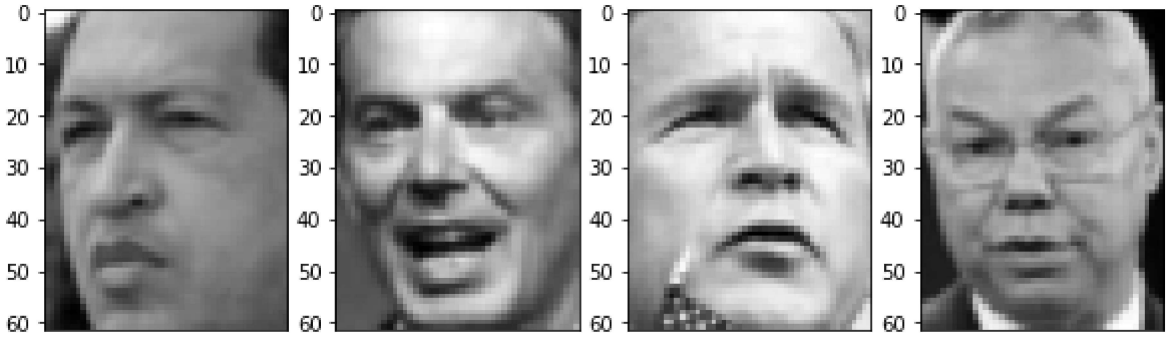
Largest eigenvalue = 12.97881887

d

$$\begin{bmatrix} 0.26018674 & -1.41900435 \\ -0.87353472 & 4.03721245 \\ -4.04749635 & -1.84867733 \\ 4.66084433 & -0.7695308 \end{bmatrix}$$

In [19]:
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import fetch_lfw_people
lfw_people = fetch_lfw_people(min_faces_per_person=70)
n_samples, h, w = lfw_people.images.shape
print(lfw_people.images.shape)
npix = h*w
fea = lfw_people.data
def plt_face(x):
    global h,w
    plt.imshow(x.reshape((h, w)), cmap=plt.cm.gray)
    plt.xticks([])
plt.figure(figsize=(10,20))
nplt = 4
for i in range(nplt):
    plt.subplot(1,nplt,i+1)
    plt_face(fea[i])
plt.show()
```

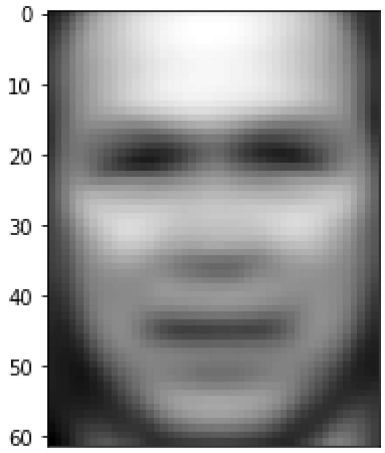
```
(1288, 62, 47)
```



In [31]:
```
#Question 5 (a)
plt_face(fea[3])
plt.show()
```

In [24]:
```
#Question 5 (b)
y=np.mean(fea,axis=0)
print(y.shape)
plt_face(y)
plt.show()
```

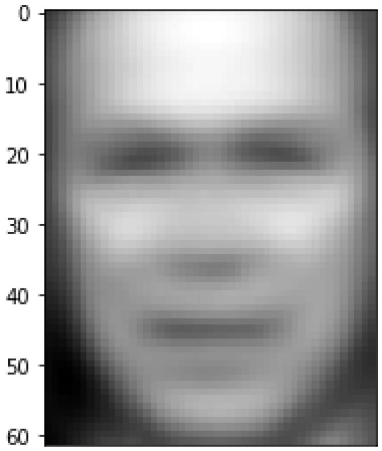
```
(2914,)
```



In [82]:
```
#Question 5 C
import sklearn.decomposition as skd
import numpy as np
X=fea-y
pca = skd.PCA(n_components = 5)
skd.PCA.fit(pca,X)
W1 = pca.components_
W = W1.transpose()
Z = pca.transform(X)
print(Z[3,])
```

```
[ 202.54202  -261.47684   418.97412   -29.39988    39.783478]
```

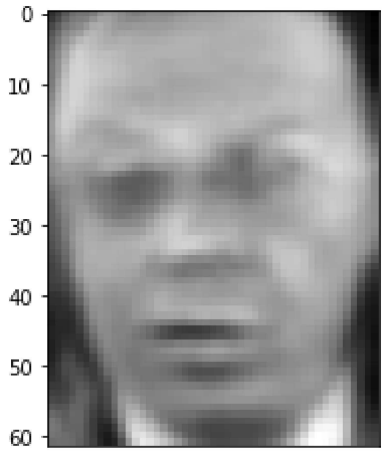
In [83]:
```
#Question 5 (d)
# for components = 5
XX=np.matmul(Z,W.transpose())
XX+=y
plt_face(XX[3])
plt.show()
```



In [79]:
```
#Question 5 (d)
# for components = 50
import sklearn.decomposition as skd
import numpy as np
X=fea-y
pca = skd.PCA(n_components = 50)
skd.PCA.fit(pca,X)
W1 = pca.components_
W = W1.transpose()
Z = pca.transform(X)
print(W1.shape)
print(Z.shape)

XX=np.matmul(Z,W.transpose())
XX+=y
plt_face(XX[3])
plt.show()
```

```
(50, 2914)
(1288, 50)
```



In [ ]:

In [ ]: